Conhecendo o editor

CARSON FIRE

Vim

neovim

Eustáquio Rangel

Twitter: @taq

E-mail: taq@eustaquiorangel.com

Uso Vim até no banheiro!
ORIGINAL
Vim
CLORO GEL

História

O editor Vim foi criado por Bram Moolenaar em 1991 para a plataforma Amiga, como um clone do editor Vi, que era utilizado em Unix e foi criado por Bill Joy em 1979. Em 1992 foi liberada uma versão do Vim para a plataforma Unix.

Bram Moolenar faleceu em Agosto de 2023, nos deixando um legado enorme.

O editor Vi

Foi um dos primeiros editores que introduziu o conceito de diferentes modos para tarefas diferentes:

- Editar texto
- Selecionar texto
- Executar comandos

Bill Joy

Em 1982, Joy co-fundou a Sun Microsystems.

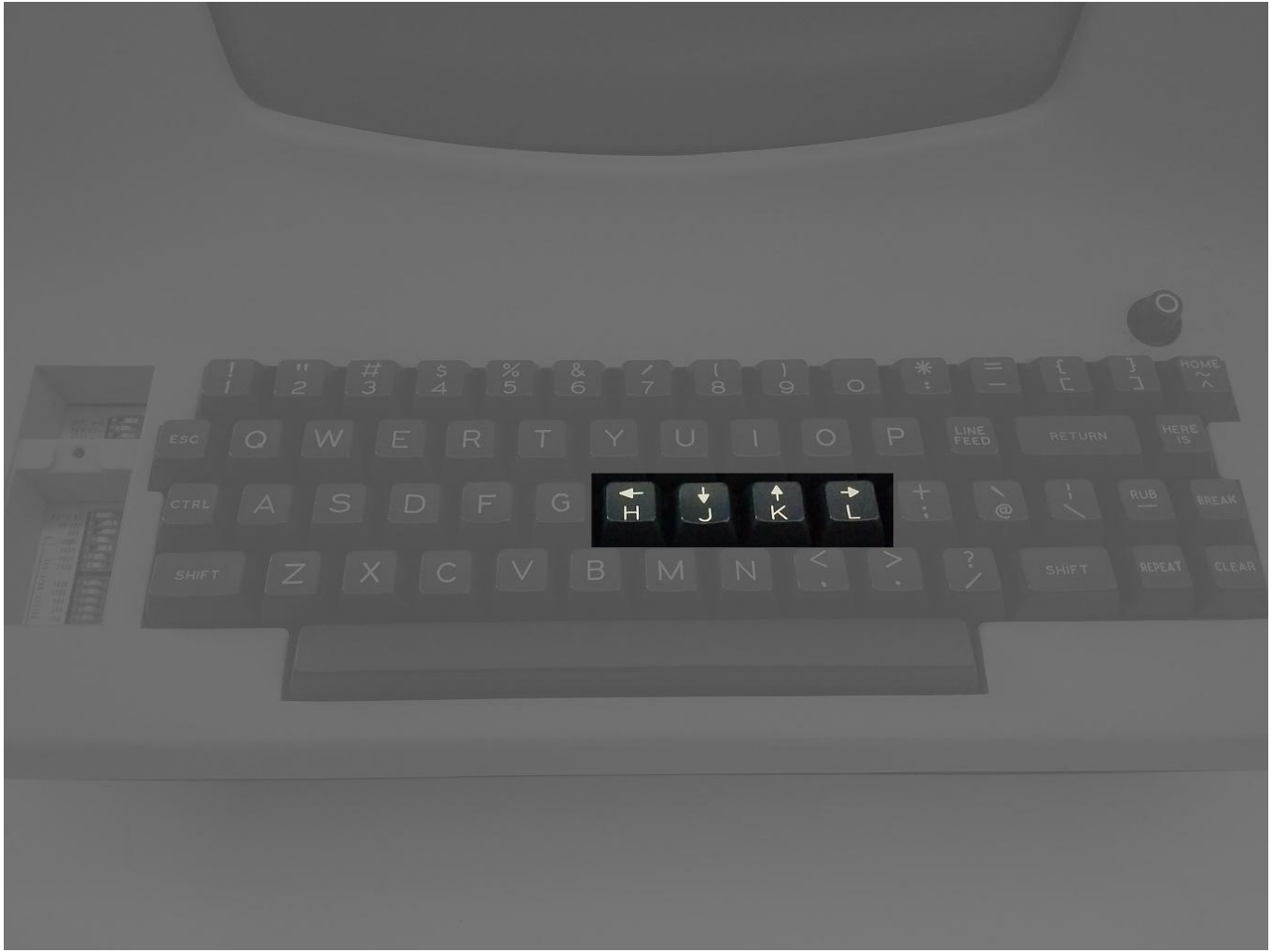
Em 1986, Joy recebeu o Prêmio Grace Murray Hopper pela ACM pelo seu trabalho no sistema operacional Berkeley UNIX.

Reza a lenda que ele ...

- Escreveu o editor Vi em um fim de semana.
- Escreveu o kernel do BSD sozinho ... em um fim de semana.
- Pelo jeito, ele não tinha muita vida social no fim de semana. #foreveralonedev

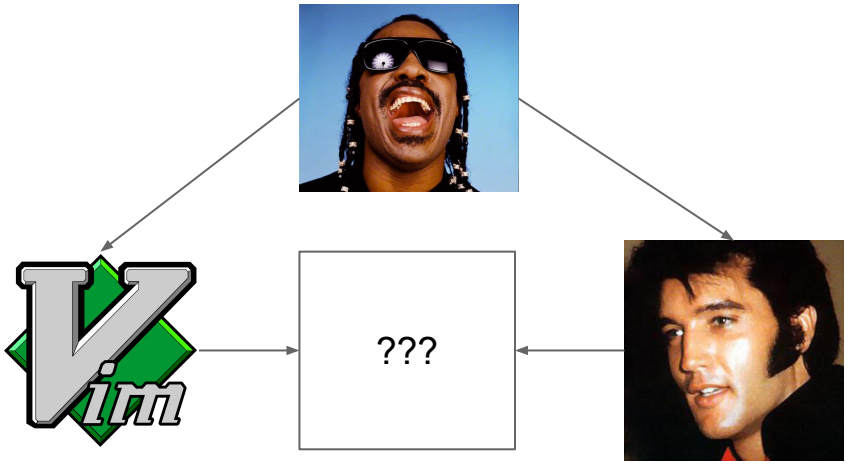
ADM-3A

Lançado 1976 pelo singelo preço de US$ 995, mostrava 12 linhas de 80 caracteres em uma tela de 12”, onde cada caracter tinha uma matriz de 5x7 nas cores âmbar, verde ou branco, com teclado de 59 teclas. Não suportava limpar até o final da linha ou da tela.


ESC
CTRL
SHIFT
LINE FEED
RETURN
HERE IS
RUB
BREAK
SHIFT
REPEAT
CLEAR
H
J
K
L

Parentesco do Vim

A primeira versão do Vim era, assim como o editor Elvis, baseada no editor Stevie.



https://neovim.io/charter/

O que é Neovim?

Neovim é um editor de texto baseado em Vim projetado para extensibilidade e usabilidade, para incentivar novas aplicações e contribuições.

Metas

Extensível. Utilizável. Vim.

- Mantenha o estilo do Vim: rápido, versátil e quase mínimo.
- Habilite novos colaboradores, remova barreiras de entrada.
- Desbloquear autores de plugins.
- Entregue uma interface Lua de primeira classe, como alternativa ao Vimscript.
- Aproveite o desenvolvimento contínuo do Vim.
- Otimize "pronto para usar", para novos usuários, mas especialmente para usuários regulares.
- Ofereça uma experiência consistente entre plataformas, visando todas as plataformas suportadas pela libuv.
- Em questões de gosto/ambiguidade, privilegie a tradição/compatibilidade...
- ...mas prefira a usabilidade se os benefícios forem extremos.

Não-objetivos

- Suporte Vim9script
- Transformar o Vim em um IDE
- Limitar aplicativos de terceiros (como IDEs!) criados com Neovim
- Descontinuar Vimscript
- Conformidade com POSIX vi

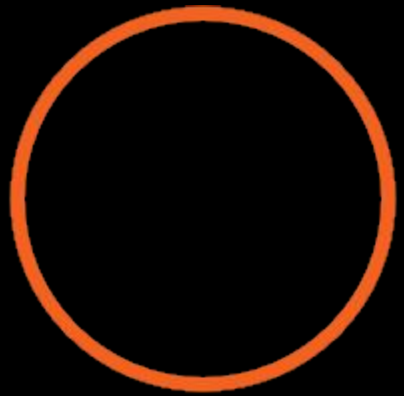
Ferramenta afiada

No episódio 40 de “Ruby Rogues”, Gary Bernhardt diz o seguinte:

> “O Vim é uma motosserra, e a última coisa que você quer em uma motosserra são os controles que não são os controles que você está acostumado, porque você pode cortar sua mão fora.”

É argumentado que alguns editores são consistentes com as expectativas, por exemplo, do sistema operacional, sem surpresas, como o TextMate no OSX, é um perfeito círculo.

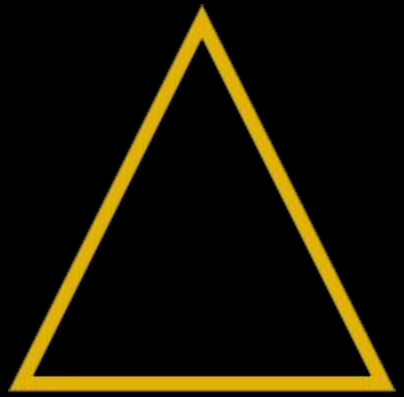
http://rubyrogues.com/040-rr-text-editors-and-ides/
http://vimcasts.org/blog/2012/08/on-sharpening-the-saw/



TextMate

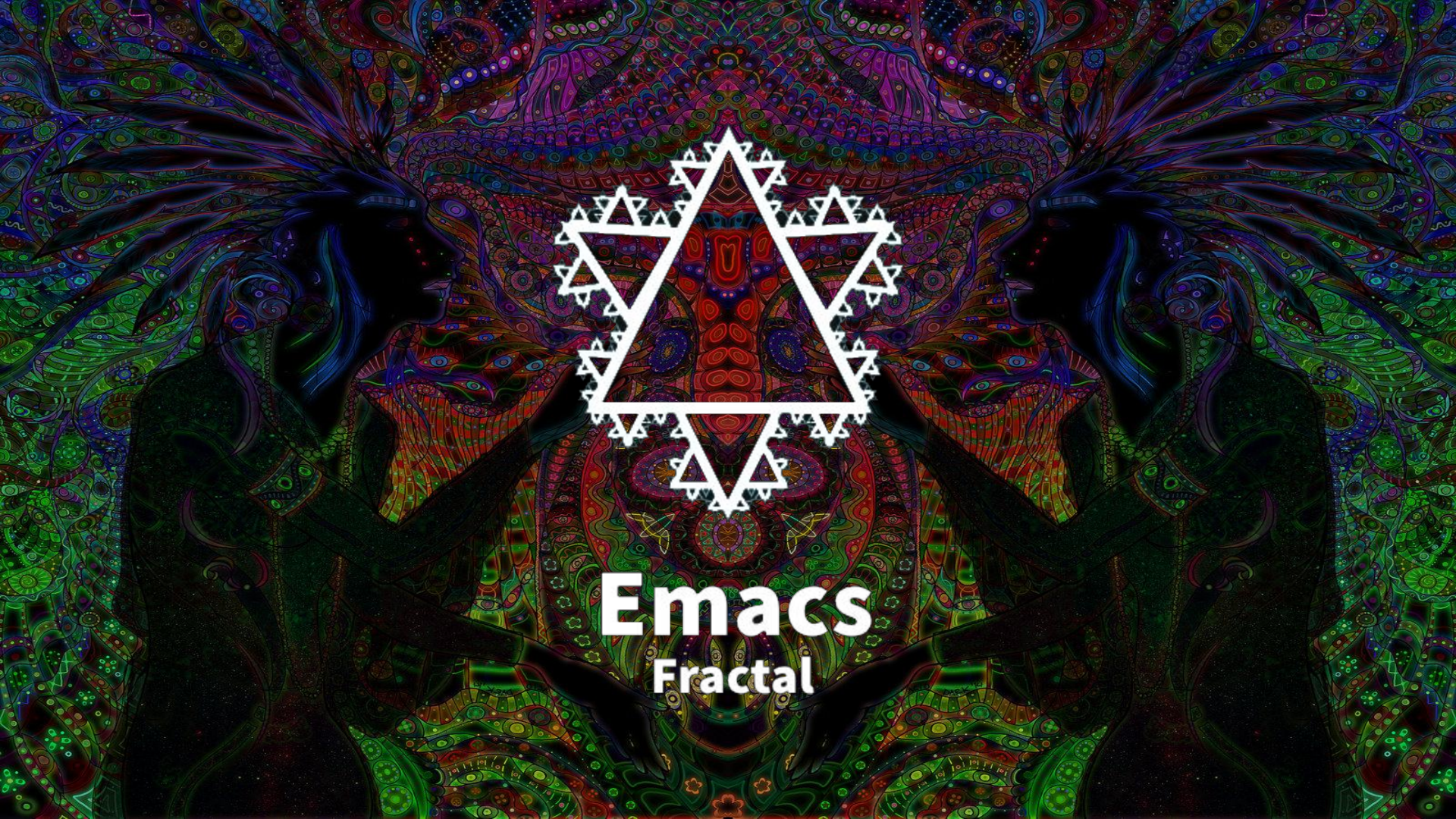
Circle

É um cílculo!



Vim

Triangle


Emacs
Fractal

Instalando

Repositórios padrão

```
$ sudo apt install vim
$ sudo apt install neovim
```

PPA

```
$ sudo add-apt-repository ppa:neovim-ppa/stable -y    # stable
$ sudo add-apt-repository ppa:neovim-ppa/unstable -y  # unstable
$ sudo apt update
$ sudo apt install neovim
```

Vamos precisar para os plugins

```
$ sudo apt install git curl wget ripgrep exuberant-ctags pdftotext
```

Modos do Vim

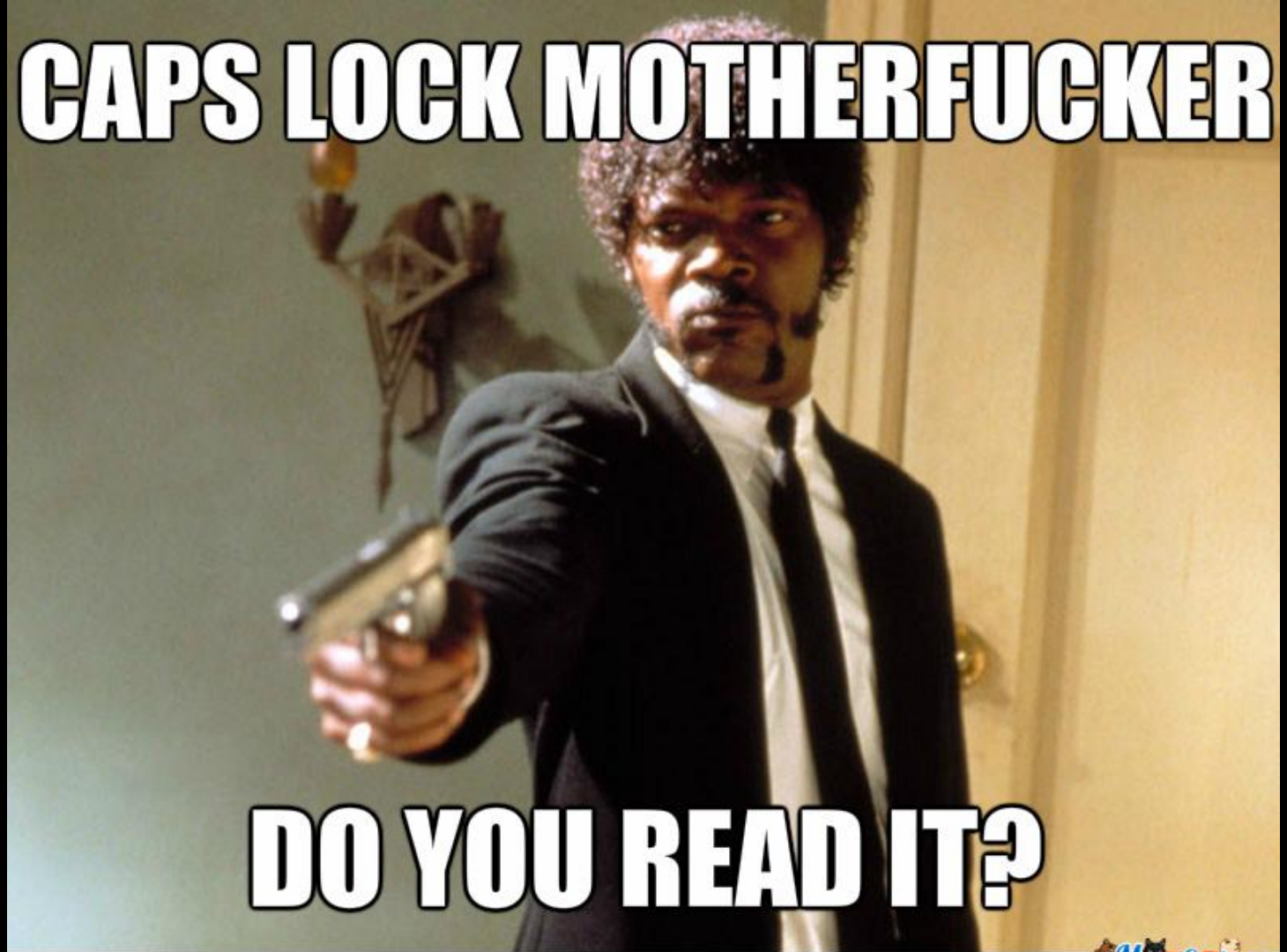
O Vim tem alguns modos:

- **Normal**, para deletar, copiar, formatar, etc
- **Inserção**, para digitação de texto
- **Visual**, para seleção de linhas, blocos
- **Comando**, para digitar comandos para o editor

Quando é aberto, o editor entra automaticamente no modo **normal**. É daí de onde vem todos os xingamentos de quem entra nele e não sabe mais como sair.

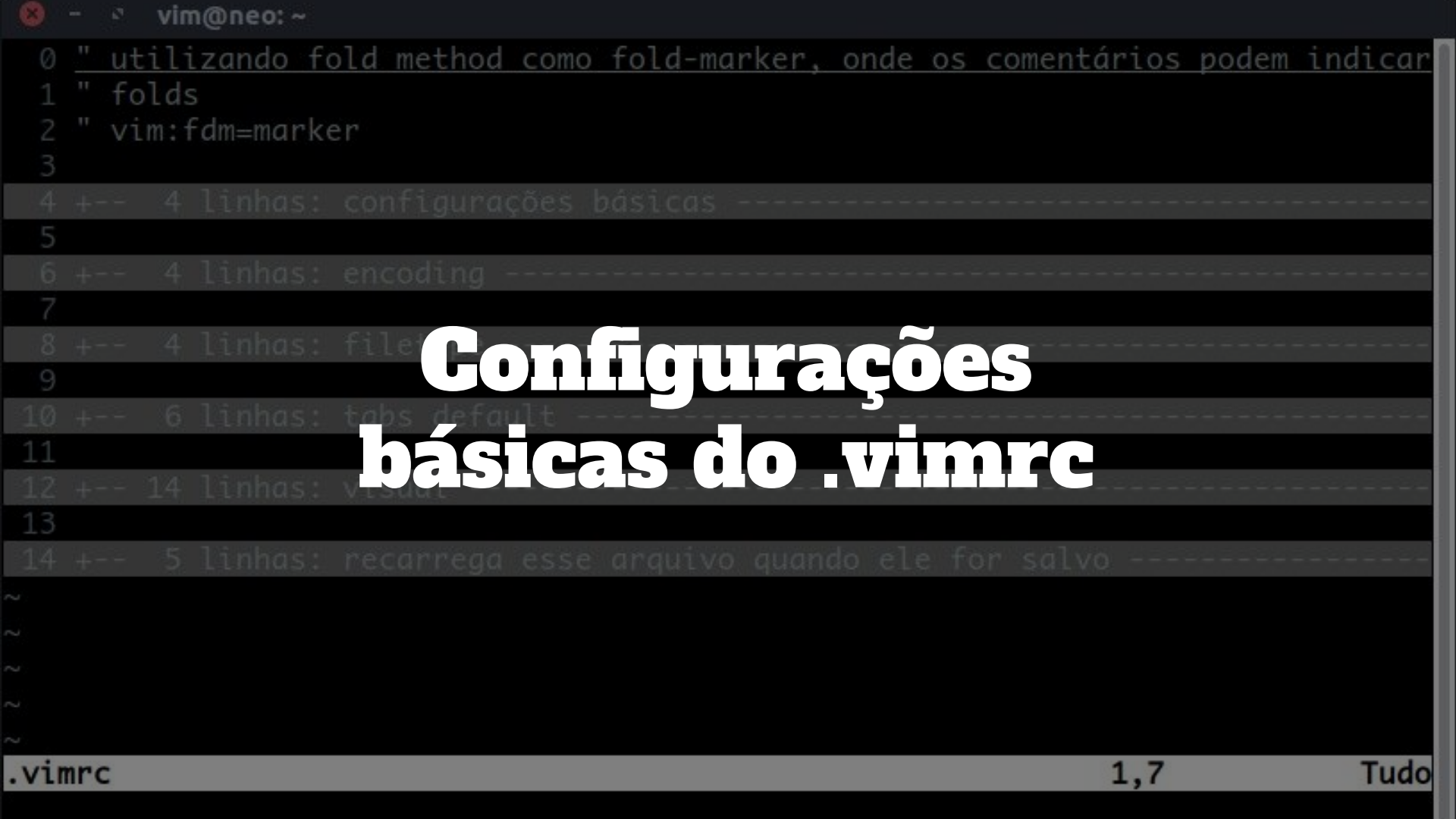
Esse cara não sabia que podia digitar **ESC** e depois **:qa!**

Perdido em algum modo? Digite ESC e "i" para entrar em modo de inserção.


CAPS LOCK MOTHERFUCKER
DO YOU READ IT?

Principais configurações

- Arquivo **.vimrc/init.vim**, my precious!
- Diretório **.vim**, onde você com certeza vai entupir de coisas e depois vai tirar um monte.


vim@neo: ~
" utilizando fold method como fold-marker, onde os comentários podem indicar
" folds
" vim:fdm=marker
+-- 4 linhas: configurações básicas
+-- 4 linhas: encoding
+-- 4 linhas: file
+-- 6 linhas: tabs default
+-- 14 linhas: visual
+-- 5 linhas: recarrega esse arquivo quando ele for salvo
Configurações básicas do .vimrc
.vimrc
1,7
Tudo

Na primeira vez, usem outro editor

Criem o seu **.vimrc** em outro editor e salvem no seu diretório **home** (**/home/<usuário>** ou **~**) para já abrir o Vim com alguma coisa configurada. Os arquivos desse workshop vão ser disponibilizados em https://github.com/taq/workshop-vim, essa versão vai estar na tag **v1**.

Podemos criar um link para o arquivo inicial do Neovim:

```
$ mkdir -p ~/.config/nvim/
$ ln -s ~/.vim ~/.config/nvim/init.vim
```

```
" utilizando fold method como fold-marker, onde os comentários podem indicar
" folds
" vim:fdm=marker

" configurações básicas {{{
set nocompatible    " não precisamos ser totalmente compatíveis com o Vi!
syntax on           " habilita syntax highlight
" }}}

" encoding {{{
set encoding=utf-8       " encoding utilizado dentro do editor
set fileencoding=utf-8   " encoding utilizado para os arquivos
" }}}

" filetype {{{
filetype plugin on     " carrega plugins do filetype
filetype indent on     " habilita indentação do filetype
" }}}
```

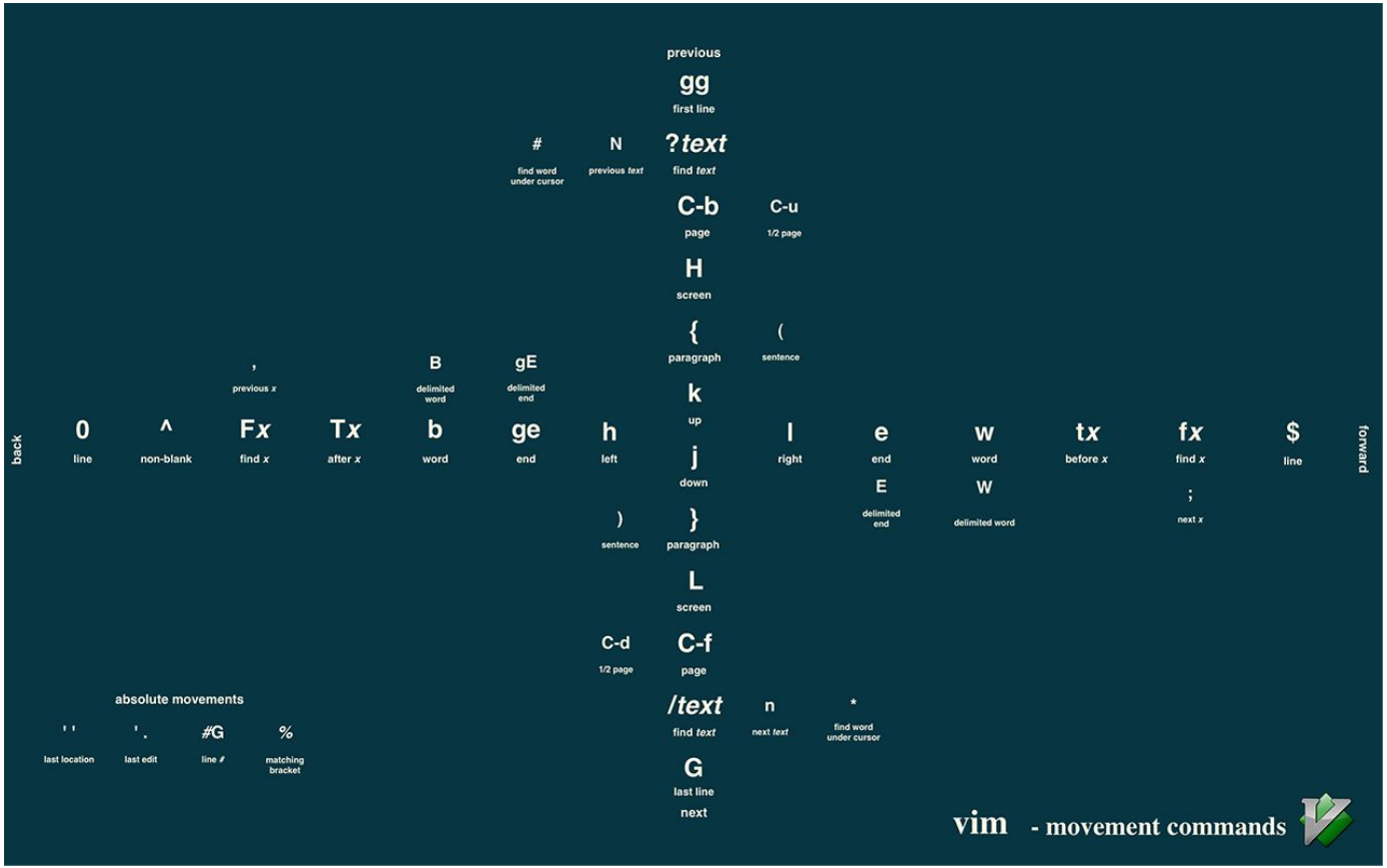
v1

```
" tabs default {{{
set tabstop=3               " número de espaços da tabulação
set softtabstop=3           " número de espaços de tabulação para edição
set shiftwidth=3            " número de espaços para usar com a indentação automática
set expandtab               " fim da guerra: use espaços ao invés de tabulações
" }}}

" visual {{{
set number relativenumber   "  número relativo da linha, salva a pátria!
set novisualbell                " sem ficar piscando, enche o saco
set smartindent             " indenta linhas novas
set foldmethod=indent       " método de fold (dobras) baseado em indentação
set ignorecase              " ignora maiúsculas e minúsculas
set hlsearch                " destaca os termos de pesquisa que foram encontrados
set incsearch               " mostra o termo de pesquisa enquanto ele é digitado
set laststatus=2                " sempre usar linhas de status
set textwidth=80                " comprimento máximo do texto inserido antes de quebrar a
linha
set cursorline              " destaca a linha que o cursor está
set showmatch               " mostra os parenteses que 'casam'
set splitbelow              " posiciona janelas divididas sempre abaixo da atual
" }}}
```

v1

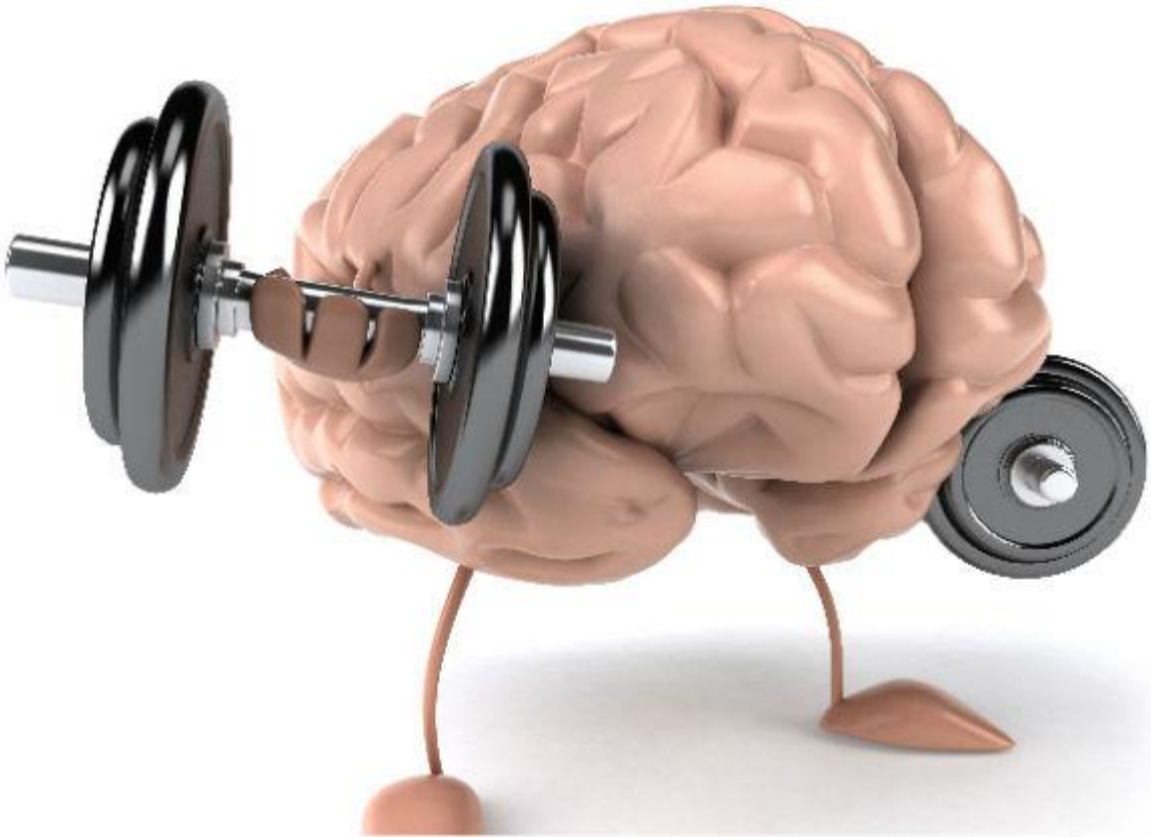
Movimentando por aí

Em modo **normal** (aperte ESC se estiver na dúvida)



https://github.com/LevelbossMike/vim_shortcut_wallpaper

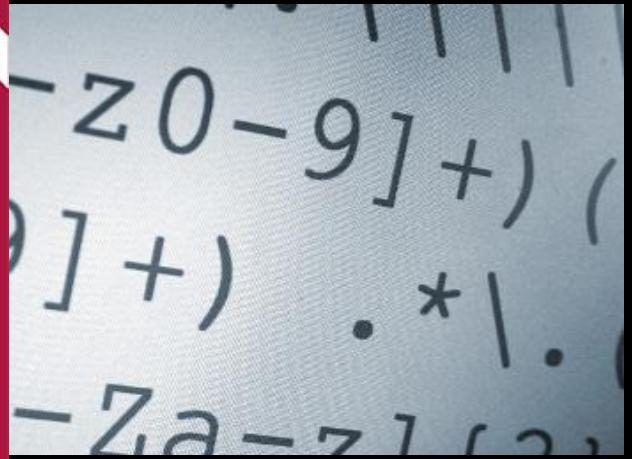
Memória muscular



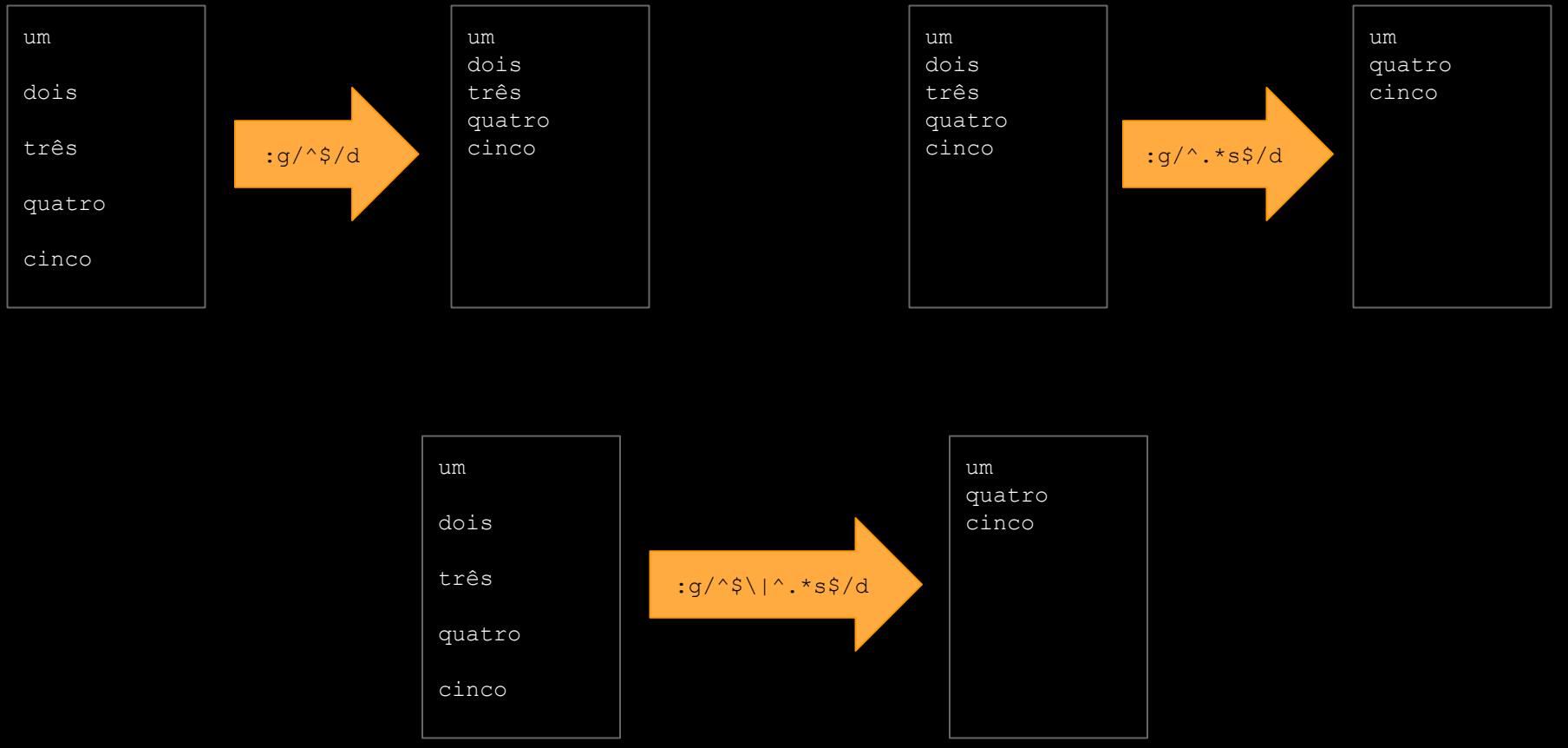
Para treinar: https://vim-adventures.coma

O comando G

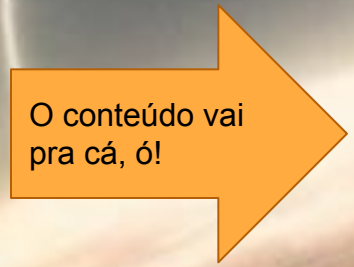
Precisamos de expressões regulares



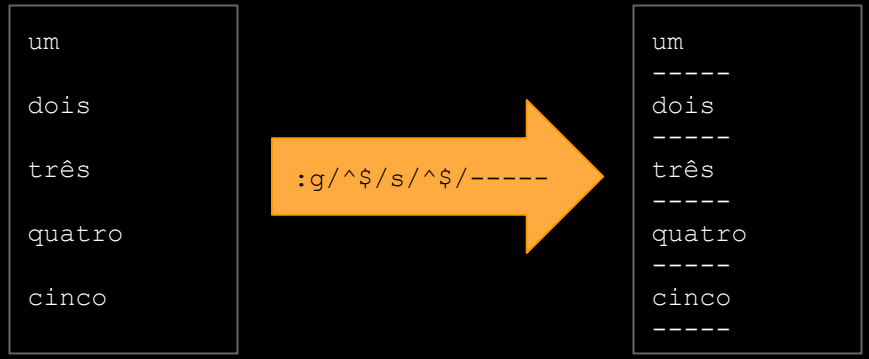
http://piazinho.com.br/
@oreio

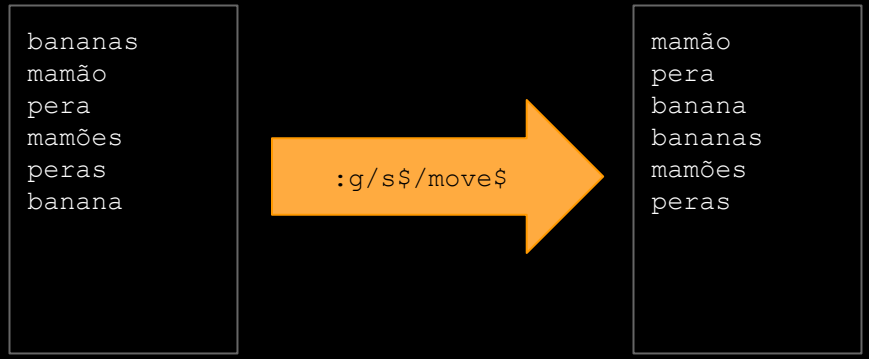

um
dois
três
quatro
cinco
:g/^$/d
um
dois
três
quatro
cinco
um
dois
três
quatro
cinco
:g/^.*s$/d
um
quatro
cinco
um
dois
três
quatro
cinco
:g/^$\|^.*s$/d
um
quatro
cinco

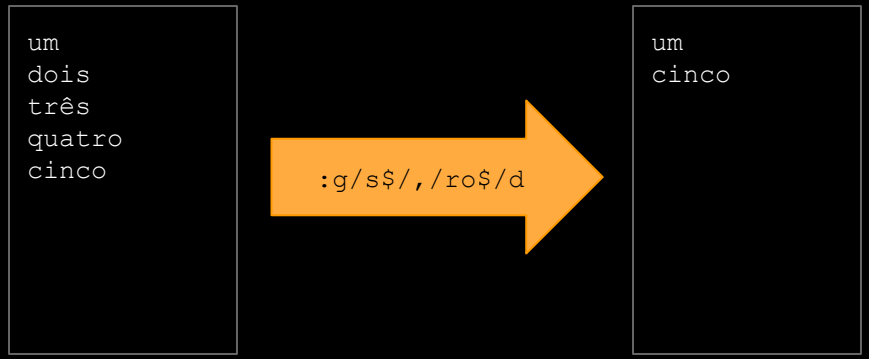
```
:g/^$/d_
```



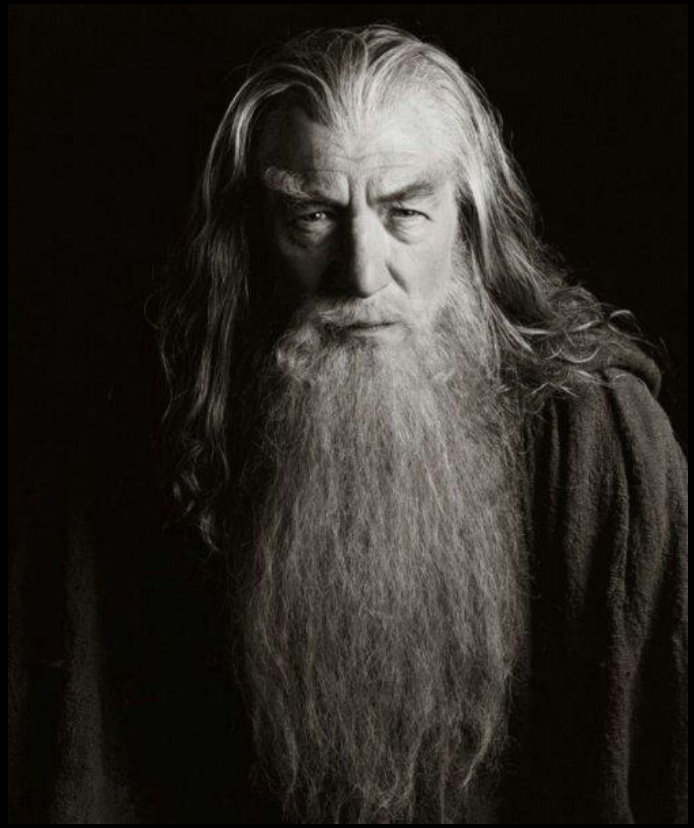
O registrador do buraco negro


um
dois
três
quatro
cinco
:g/^$/s/^$/-----
um
-----
dois
-----
três
-----
quatro
-----
cinco


bananas
mamão
pera
mamões
peras
banana
:g/s$/move$
mamão
pera
banana
bananas
mamões
peras


um
dois
três
quatro
cinco
:g/s$/,/ro$/d
um
cinco

Falando sobre expressões regulares ...



Magic mode!

Utilizando **\v** antes de qualquer expressão regular, não precisa “escapar” os caracteres, como por exemplo, para trocar o primeiro nome pelo segundo:

```
eustaquio rangel

# sem magic mode
:%s/\(\w\+\) \(\w\+\)/\2 \1/g

# com magic mode
:%s/\v(\w+) (\w+)/\2 \1/g

# yay! bem melhor!
```

Registradores

Copiar (yank)

- Linha corrente: **yy**
- Seleção corrente: **y**

Colar (paste)

- Conteúdo corrente: **p**

Copiar para registrador (yank)

- Linha corrente: **<ESC>”<registrador>yy**
- Seleção corrente: **<ESC>”<registrador>y**

Colar (paste)

- Conteúdo corrente: **<ESC>”<registrador>p**

Visualizando os registradores: **:registers**

Gravando (macros)

Começar a gravar: <ESC>q<registrador>
Parar a gravação: <ESC>qq
Repetir a gravação: <ESC>@<registrador>
Repetir a gravação várias vezes: <ESC><número>@<registrador>

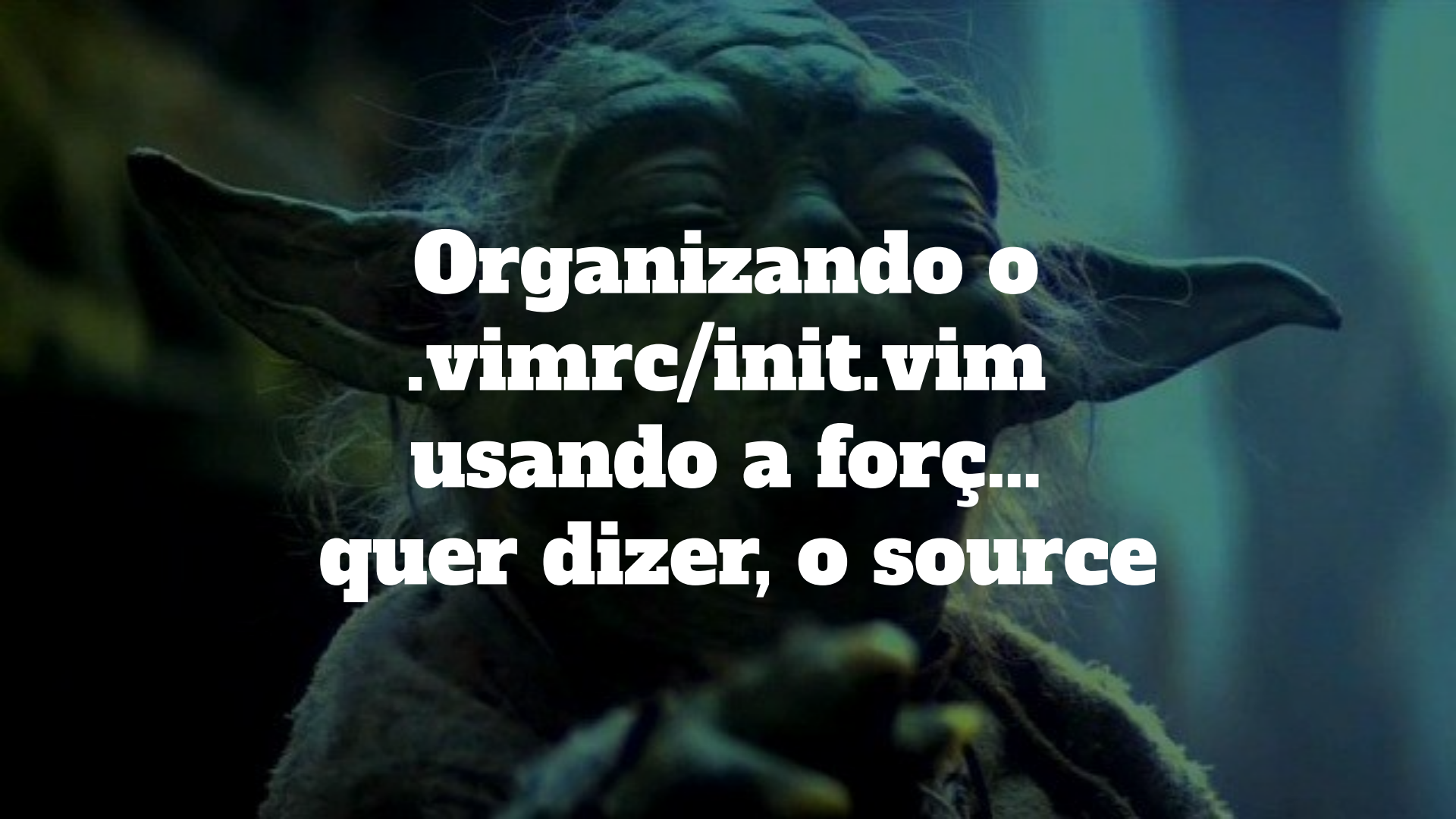
```
1 um
2 dois
3 três
4 quatro
5 cinco
```

qa
@5a

```
h = {
1 => 'um',
2 => 'dois',
3 => 'três',
4 => 'quatro',
5 => 'cinco'
}
```

Navegando no terminal

```
# .bashrc
set -o vi
```


Organizando o
.vimrc/init.vim
usando a forç...
quer dizer, o source

.vimrc

source ~/.vim/custom/gui.vim

/.vim/custom/gui.vim

```
" mostra caracteres que ficam escondidos:
" tab: dois caracteres, um para a tab e outro para ocupar o espaço da tab
" trail: caracter para espaços extras (sobrando no final de linha)
" extends: mostra quando wrap está desligado e a linha excede o limite
" precedes: mostra quando wrap está desligado e tem texto antes
set listchars=tab:▸\ ,trail:·,extends:>,precedes:<

" mostra os caracteres definidos acima
set list

" fundo escuro
set background=dark
```

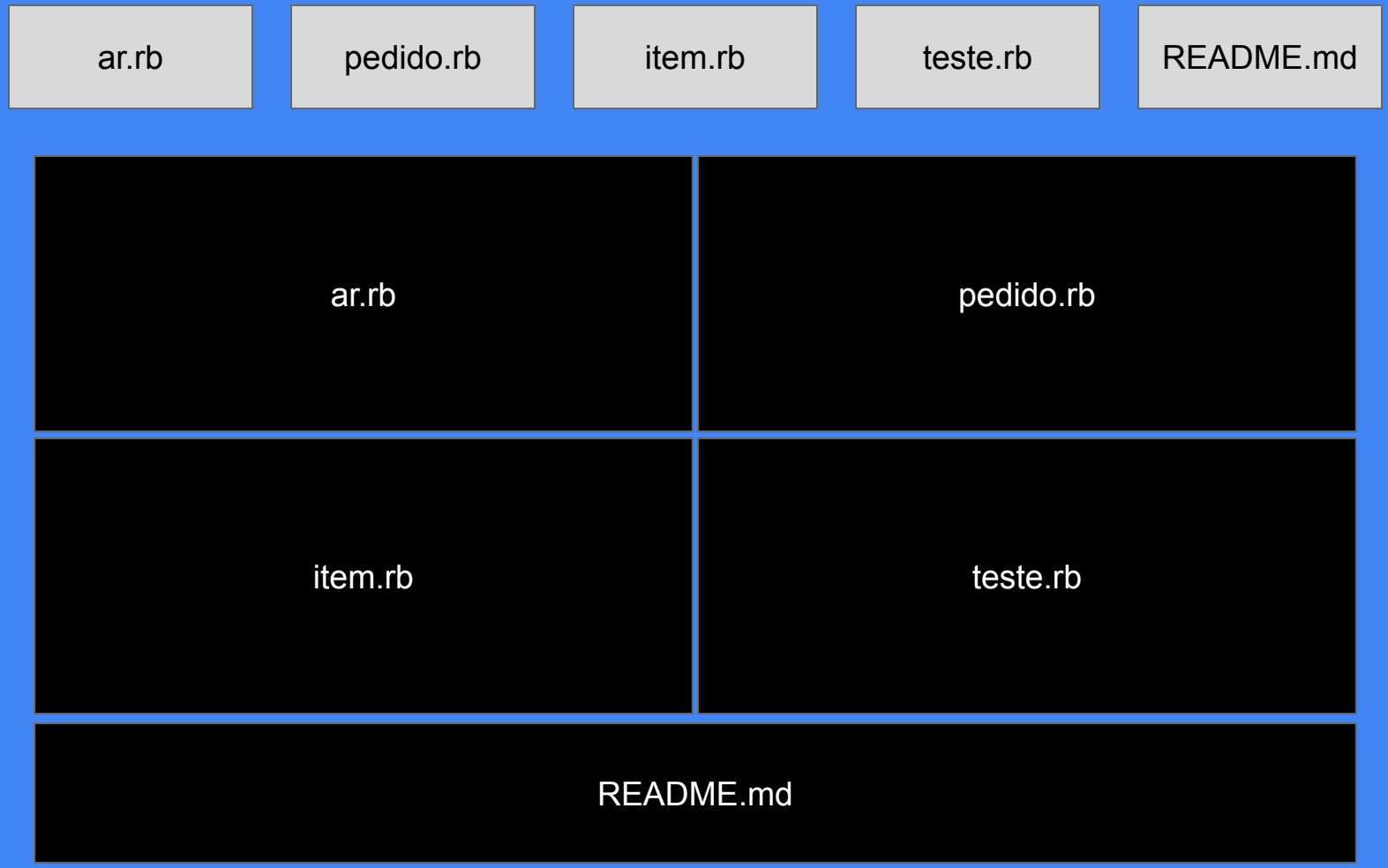
v2

Os famigerados arquivos .swp

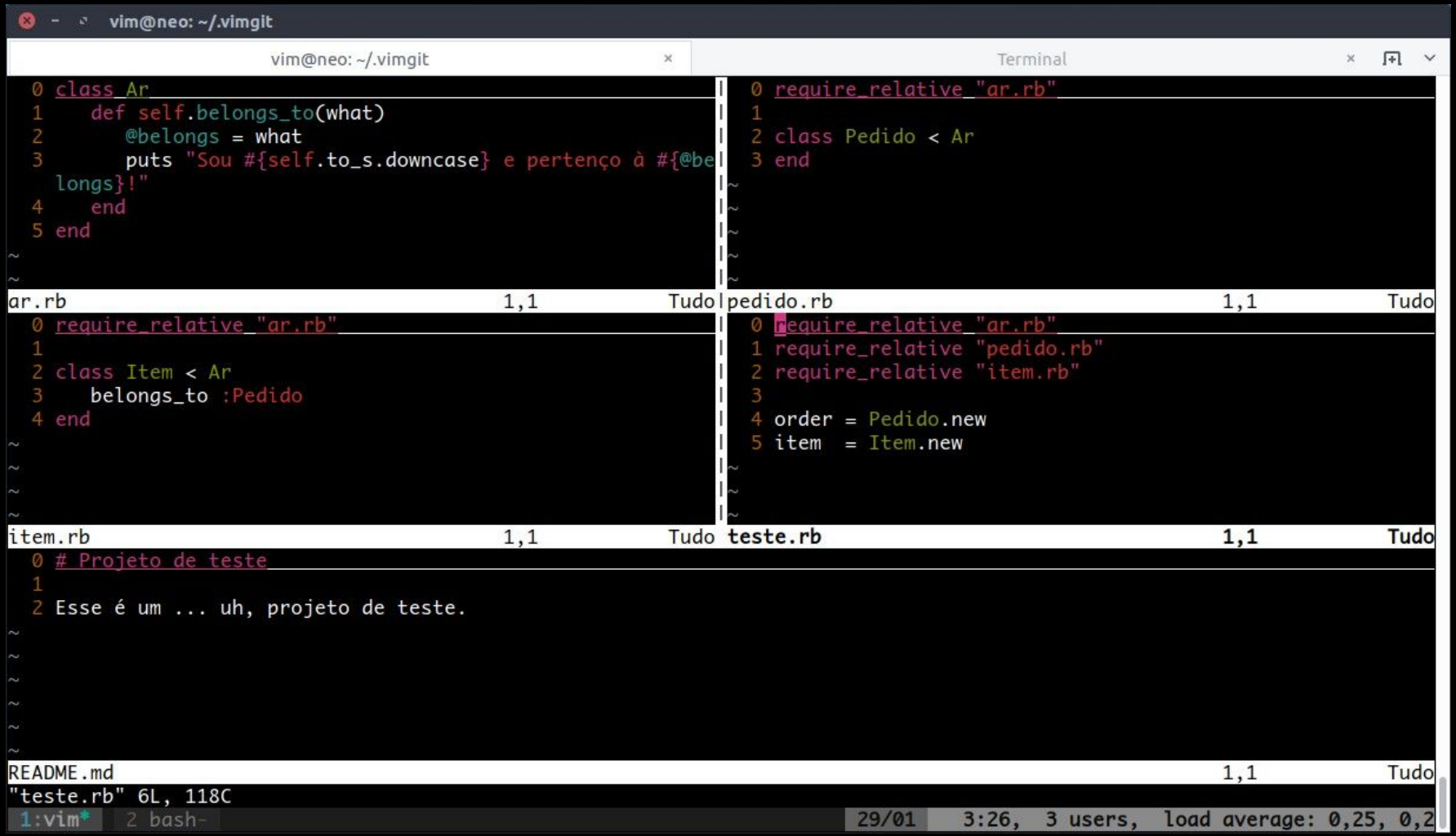
```
" backup {{{
set nobackup      " não mantém arquivos .swp
set writebackup   " cria um arquivo de backup quando sobrescreve
" }}}
```

v3

Trabalhando com vários arquivos


ar.rb
pedido.rb
item.rb
teste.rb
README.md
ar.rb
pedido.rb
item.rb
teste.rb
README.md

```
:set splitright  " pode ser colocado no .vimrc!
:e ar.rb         " abre o arquivo ar.rb na janela atual
:sp README.md    " abre o arquivo README.md abaixo
CTRL+w k         " move para a janela acima
:vs pedido.rb    " abre o arquivo pedido.rb à direita
CTRL+w h         " move para a janela à esquerda
:sp item.rb      " abre o arquivo item.rb abaixo
CTRL+w l         " move para a janela à direita
:sp teste.rb     " abre o arquivo teste.rb abaixo
CTRL+w =         " regulariza o tamanho das janelas
```


vim@neo: ~/.vimgit
vim@neo: ~/.vimgit
Terminal
0 class Ar
1 def self.belongs_to(what)
2 @belongs = what
3 puts "Sou #{self.to_s.downcase} e pertenço à #{@be
longs}!"
4 end
5 end
ar.rb 1,1 Tudo
0 require_relative "ar.rb"
1
2 class Pedido < Ar
3 end
pedido.rb 1,1 Tudo
0 require_relative "ar.rb"
1
2 class Item < Ar
3 belongs_to :Pedido
4 end
item.rb 1,1 Tudo
0 require_relative "ar.rb"
1 require_relative "pedido.rb"
2 require_relative "item.rb"
3
4 order = Pedido.new
5 item = Item.new
teste.rb 1,1 Tudo
0 # Projeto de teste
1
2 Esse é um ... uh, projeto de teste.
README.md 1,1 Tudo
"teste.rb" 6L, 118C
1:vim* 2 bash-
29/01 3:26, 3 users, load average: 0,25, 0,2

k
h
CTRL+w
l
j

Abas

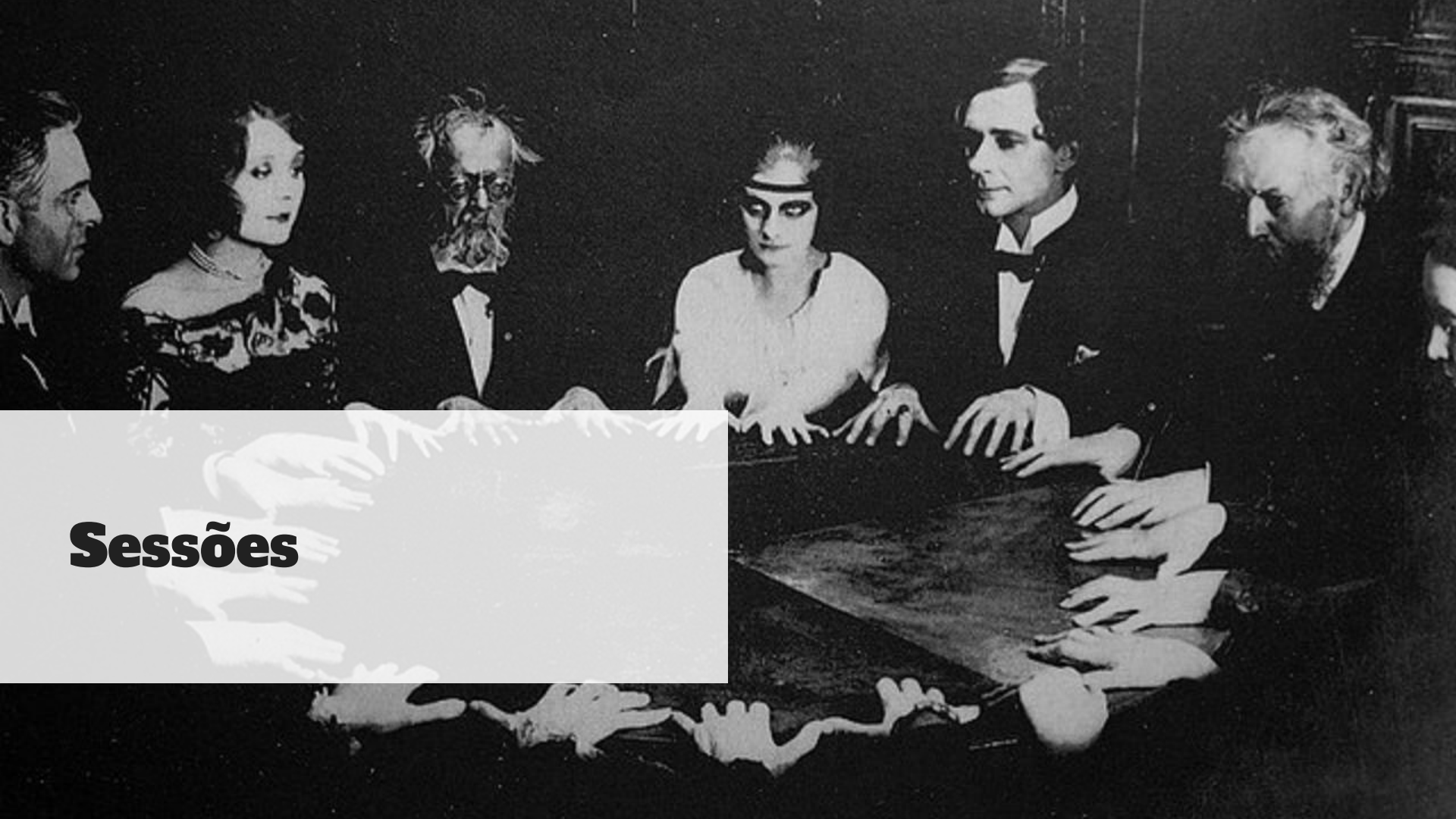
```
:e ar.rb
:tabnew pedido.rb
:tabnew item.rb
:tabnew teste.rb
:tabnew README.md
```

Em modo normal (aperte ESC):

gt - vai para a próxima aba

gT - vai para a aba anterior

Navegando pelas abas



Sessões

```
" sessões {{{
map ms :mksession! /tmp/session.vim<CR>
map rs :source /tmp/session.vim<CR>
" }}}
```

v4

Correção ortográfica

```
" correção ortográfica {{{
map spon :set spell spelllang=pt<CR>
map spof :set nospell<CR>
" }}}
```

spellfile#LoadFile(): There is no writable spell directory.
Shall I create /home/vim/.vim/spell
(Y)es, [N]o: Y

Cannot find spell file for "pt" in utf-8
Do you want me to try downloading it?
(Y)es, [N]o: Y

Downloading pt.utf-8.spl...
:!links 'http://ftp.vim.org/pub/vim/runtime/spell/pt.utf-8.spl' -source > '/tmp/v2bQwNV/0.spl'

In which directory do you want to write the file:
1. /home/vim/.vim/spell
[C]ancel, (1): 1

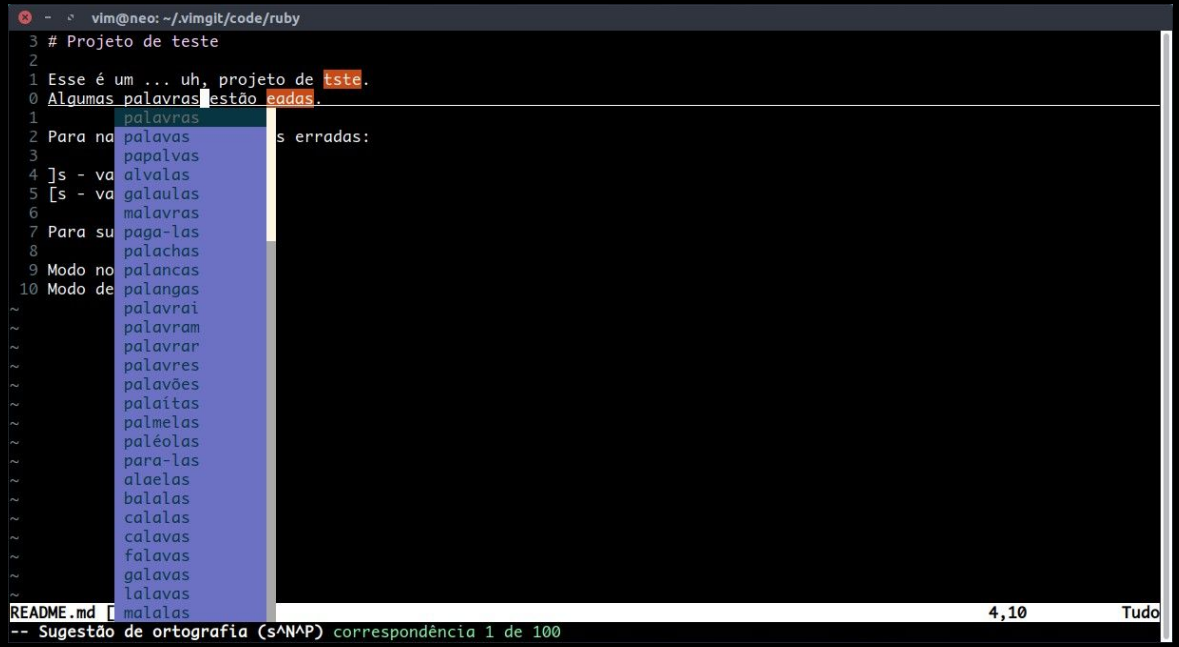
Do you want me to try getting the .sug file?
This will improve making suggestions for spelling mistakes,
but it uses quite a bit of memory.
[N]o, (Y)es: Y

Para navegar nas palavras erradas:

]s - vai para a próxima
[s - vai para a anterior

Para sugestões

Modo **normal**: z=
Modo de **inserção**: CTRL+x s

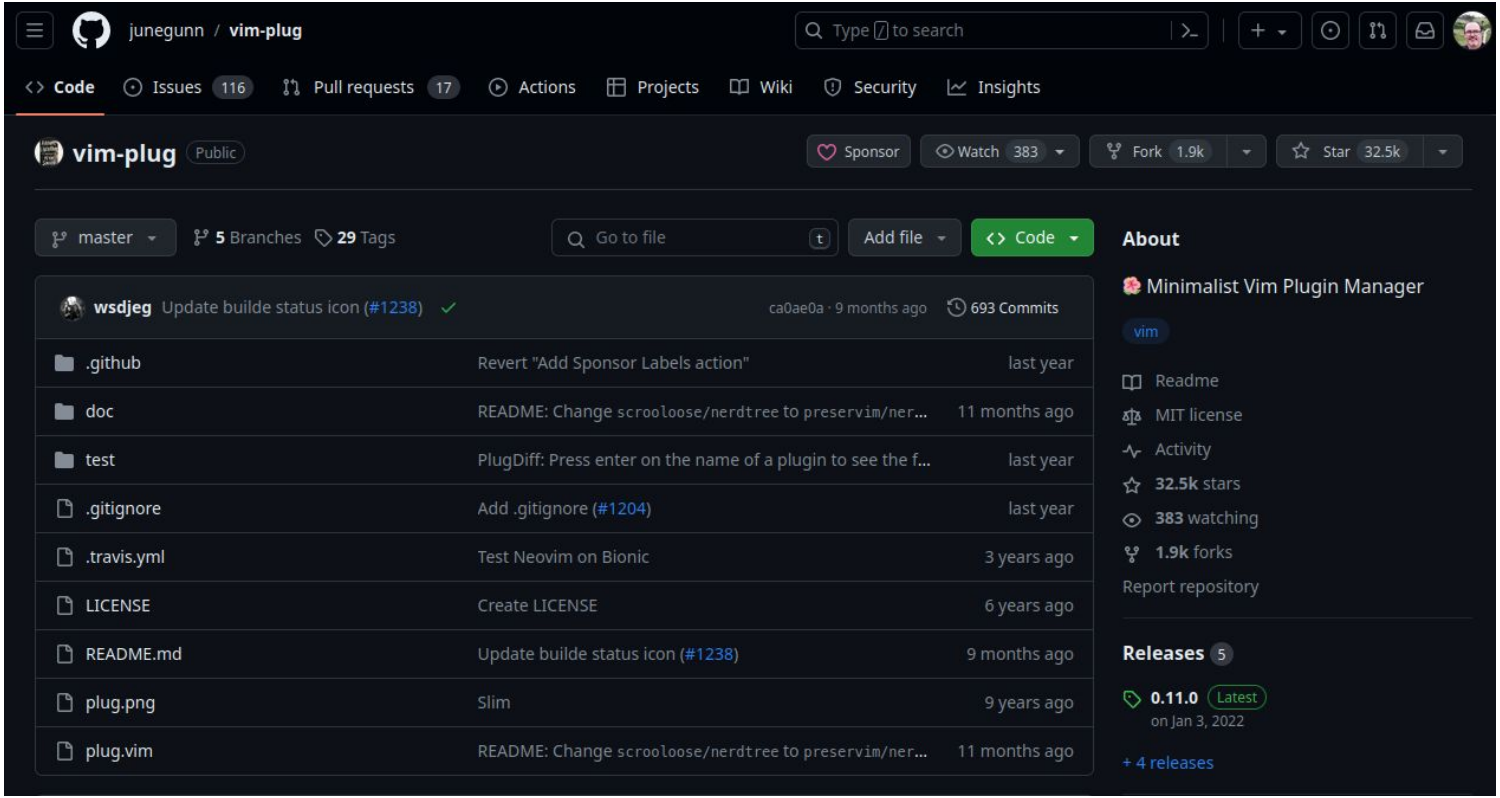


v5

Plugins

Guerra dos gerenciadores

Vim-plug!


junegunn / vim-plug
Type / to search
Code
Issues 116
Pull requests 17
Actions
Projects
Wiki
Security
Insights
vim-plug
Public
Sponsor
Watch 383
Fork 1.9k
Star 32.5k
master
5 Branches
29 Tags
Go to file
t
Add file
Code
wsdjeg Update builde status icon (#1238)
ca0ae0a · 9 months ago
693 Commits
.github Revert "Add Sponsor Labels action" last year
doc README: Change scrooloose/nerdtree to preservim/ner... 11 months ago
test PlugDiff: Press enter on the name of a plugin to see the f... last year
.gitignore Add .gitignore (#1204) last year
.travis.yml Test Neovim on Bionic 3 years ago
LICENSE Create LICENSE 6 years ago
README.md Update builde status icon (#1238) 9 months ago
plug.png Slim 9 years ago
plug.vim README: Change scrooloose/nerdtree to preservim/ner... 11 months ago
About
Minimalist Vim Plugin Manager
vim
Readme
MIT license
Activity
32.5k stars
383 watching
1.9k forks
Report repository
Releases 5
0.11.0 Latest
on Jan 3, 2022
+ 4 releases

Escolhendo uma estrutura de diretórios

```
/home/<usuário>/
+
|
+---  .vimrc
+---  .config
|     |
|     +---  nvim
|           |
|           +---  init.vim
+---  .vim
      |
      +---  custom
```

```
$ curl -fLo "$HOME/.vim/custom/plug.vim" --create-dirs
https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim

# .vimrc/init.vim - coloque bem no comecinho, *após* as tabs!
" plugin manager {{{
source ~/.vim/custom/vim-plug.vim
" }}}

# criar o arquivo ~/.vim/custom/vim-plug.vim
source ~/.vim/custom/plug.vim
call plug#begin('~/.vim/custom')

" plugins vão aqui!

call plug#end()
```

Instalando o primeiro plugin e deixando o Vim mais bonito

```
# antes do final do ~/.vim/custom/vim-plug.vim

...

" colorschemes
Plugin 'NLKNguyen/papercolor-theme'

...
:PluginInstall

# saia do Vim, entre novamente e digite:
:color PaperColor
```

```
" vim:fdm=marker

" default options {{{
set nocompatible
set wrap
syntax on
let mapleader=","
" }}}

" encoding {{{
set encoding=utf-8
set fileencoding=utf-8
" }}}

" filetype {{{
filetype plugin on
filetype indent on
" }}}

" tabs default {{{
set tabstop=3
set softtabstop=3
set shiftwidth=3
set expandtab
" }}}
```

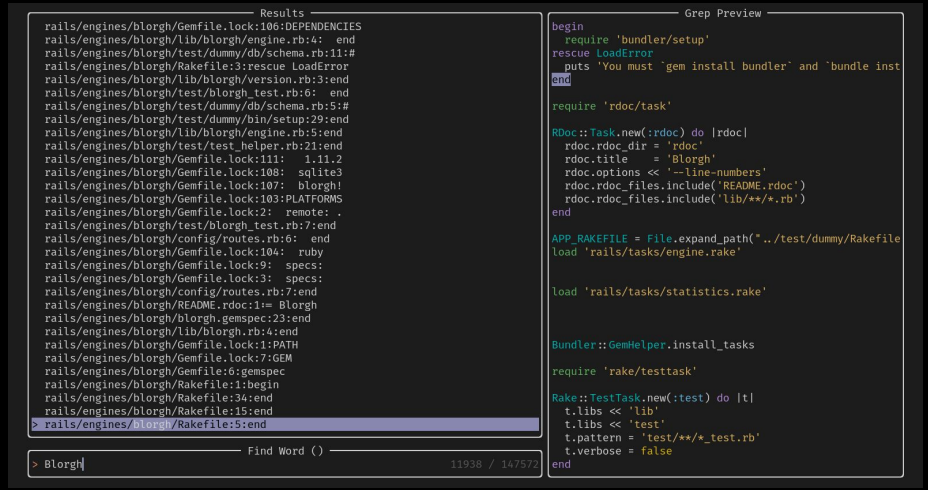
v6

Woo-Hoo!

```
:syntax on
```

Procurando as coisas

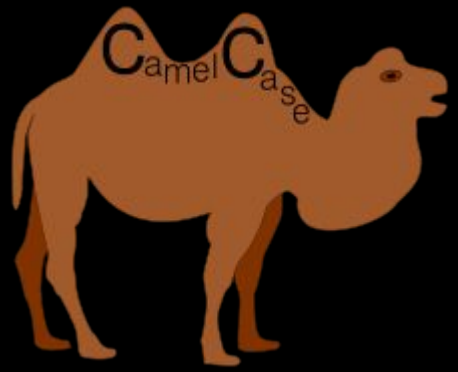
```
Plug 'nvim-lua/plenary.nvim' Plug
'nvim-telescope/telescope.nvim'

:PlugInstall
:Telescope grep_string
```



v7

Movendo aos pulinhos

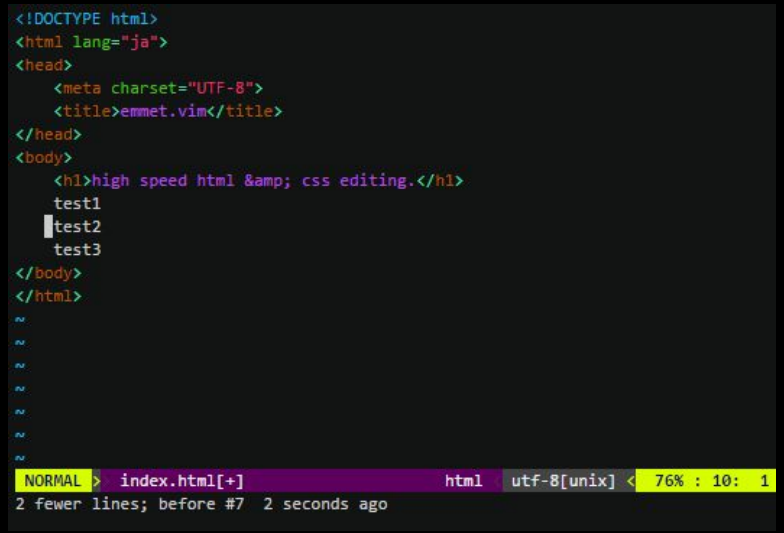
```
Plug 'vim-scripts/camelcasemotion'

:PlugInstall

# use <mapleader> <w, b, e>
```



v8

Expandindo as coisas

```
Plug 'mattn/emmet-vim.git'

:PlugInstall
```



http://emmet.io/
https://code.google.com/archive/p/zen-coding/

v9

Comentando código

```
# opcional, mas é legal
let mapleader=","

Plug 'scrooloose/nerdcommenter.git'

:PlugInstall

# use agora selecionando as linhas
# para comentar/descomentar com
# <leader>c<space>
```

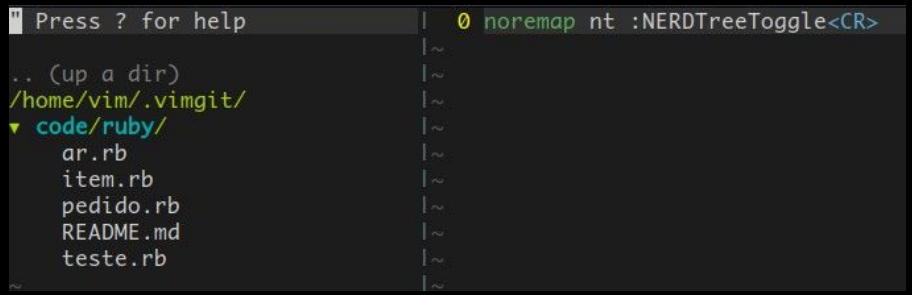
```
require_relative "ar.rb"
require_relative "pedido.rb"
require_relative "item.rb"

order = Pedido.new
item  = Item.new
```

```
require_relative "ar.rb"
require_relative "pedido.rb"
require_relative "item.rb"

#order = Pedido.new
#item  = Item.new
```

v10

Navegando

```
" programming
Plug 'scrooloose/nerdtree.git'

# ~/.vimrc
source ~/.vim/custom/nerdtree.vim

# ~/.vim/custom/nerdtree.vim
noremap nt :NERDTreeToggle<CR>

:PlugInstall

# ativem com <ESC>nt
```

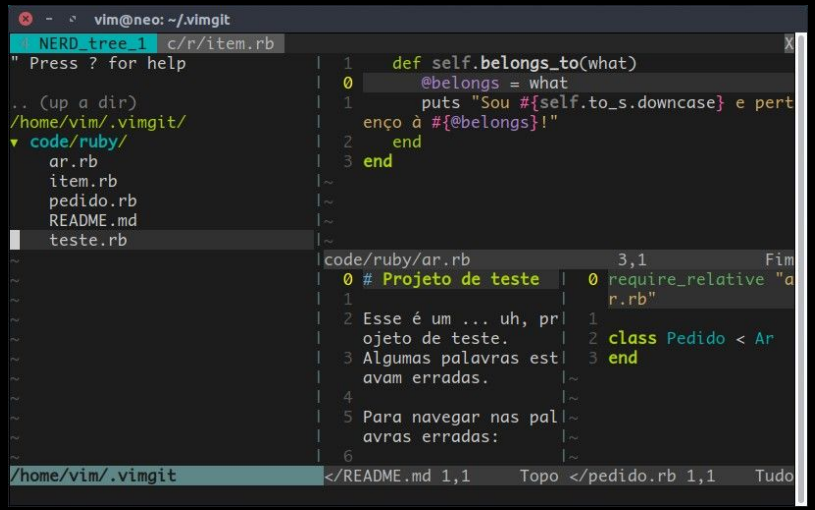


v11

Navegando

? - ajuda
o - abre ao lado
t - abre em nova aba
i - abre em novo split horizontal
s - abre em novo split vertical
r - recarrega diretório corrente

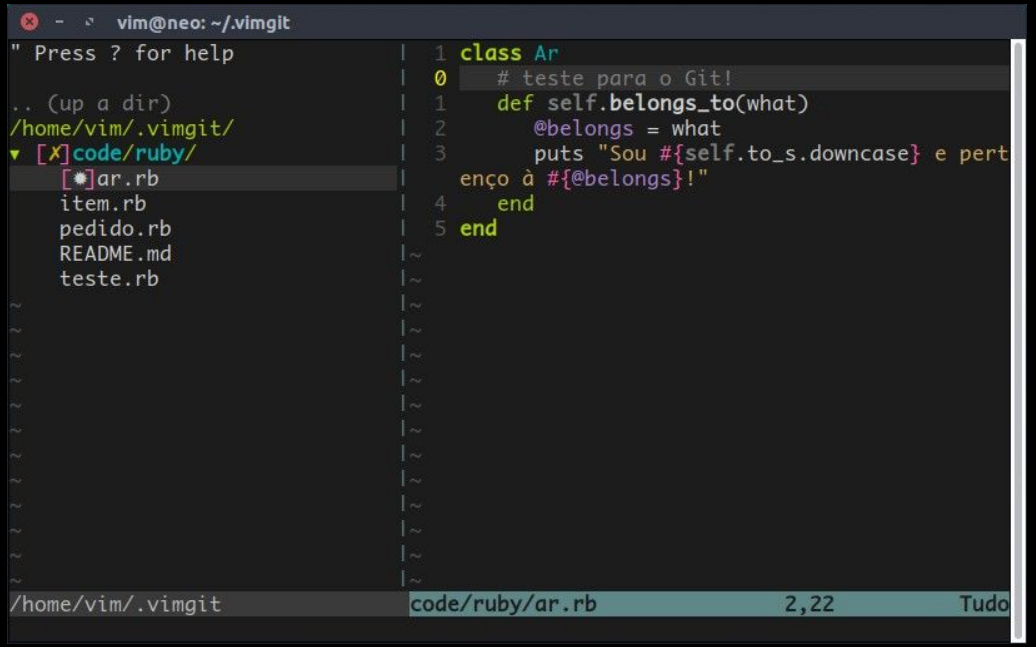
e muitas outras.



v11

Navegando com status do Git

```
# programming
Plug 'Xuyuanp/nerdtree-git-plugin'

:PlugInstall
```

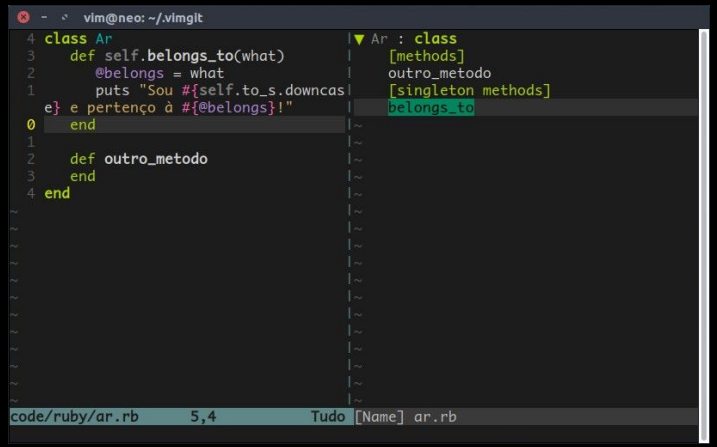


v12

Navegando em métodos e variáveis

precisa das exuberant ctags!

```
" programming
Plug 'majutsushi/tagbar'

# ~/.vimrc
source ~/.vim/custom/tagbar.vim

# ~/.vim/custom/tagbar.vim
noremap tb :TagbarToggle<CR>

:PlugInstall
```



v13

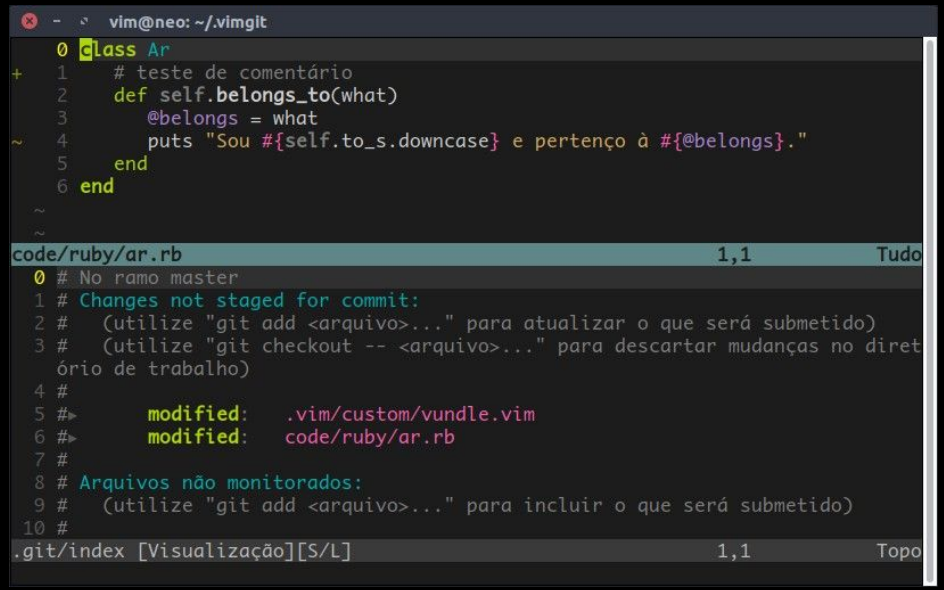
Snippets de código

```
Plug 'tomtom/tlib_vim.git'
Plug 'MarcWeber/vim-addon-mw-utils.git'
Plug 'garbas/vim-snipmate'
Plug 'honza/vim-snippets.git'

:PlugInstall
```

v14

Integração com o Git

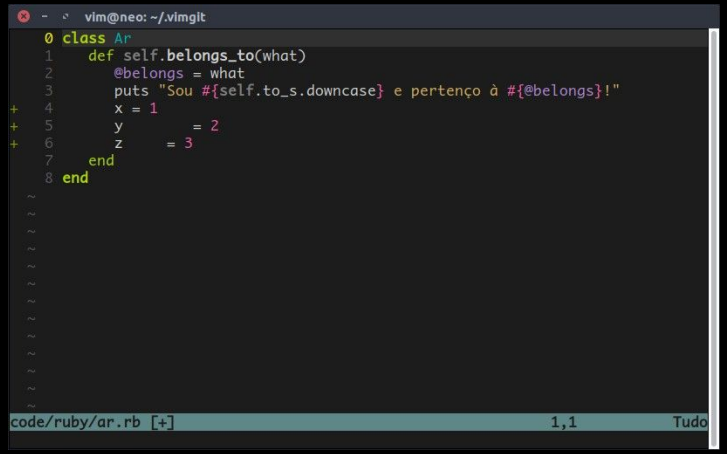
```
" programming
Plug 'tpope/vim-fugitive'
Plug 'airblade/vim-gitgutter'

:PlugInstall
```

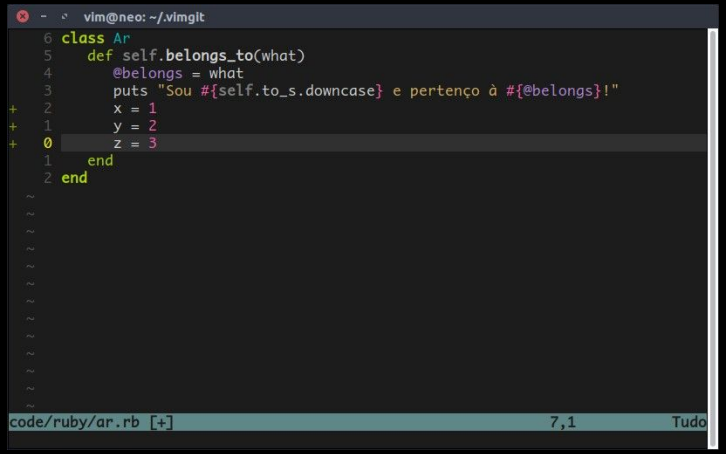


v16

Alinhando o código

```
" programming
Plug 'junegunn/vim-easy-align.git'

# .vim/custom/easyalign.vim
xmap ga <Plug>(EasyAlign)
nmap ga <Plug>(EasyAlign)

:PlugInstall
```

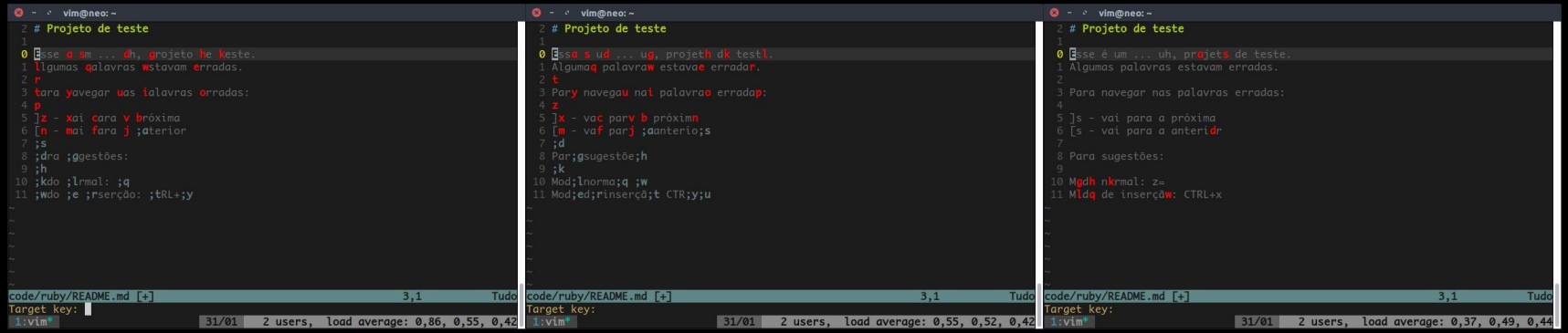




v17

Navegando nas palavras

```
" programming
Plug 'easymotion/vim-easymotion'

:PlugInstall

Utilizar com
<leader><leader><motion> ou
<leader><leader>f<caracter>
```

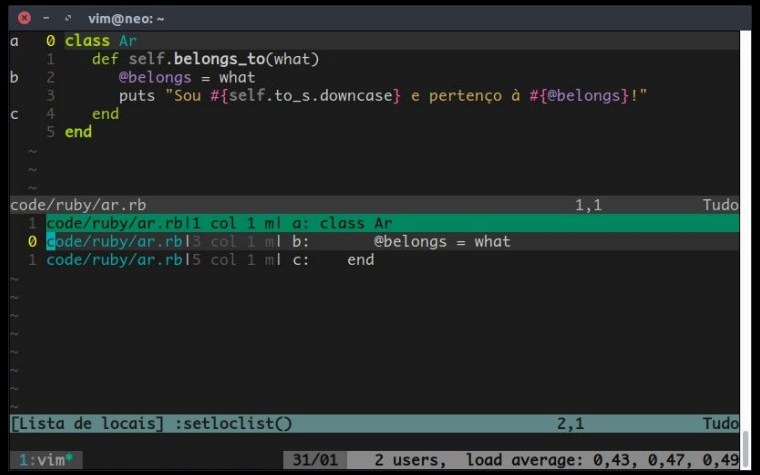


v18

Marcando seu arquivo

```
" programming
Plug 'kshenoy/vim-signature'

:PlugInstall
```

m. - Insere (e remove se tiver) próxima marca
m- - Remove a marca
m<space> - Remove todas as marcas
m/ - Abre a lista de localização do buffer corrente



v22

Comendo pelas bordas

```
" programming
Plug 'tpope/vim-surround'

:PlugInstall

# exemplos
"teste”           -> cw"'          -> 'teste'
"teste”           -> ds"           -> teste
teste             -> ysiw<em>      -> <em>teste</em> (tá, esse é chato)
<em>teste</em>    -> <esc>VS<p>    -> <p><em>teste</em></p>
```

v22

Navegando pelas tags

```
" programming
Plug 'ludovicchabant/vim-gutentags'

:PlugInstall

# navegando
CTRL+] - vai para a definição da tag abaixo do cursor
CTRL+o - retorna para onde estava

# forçando o arquivo de tags
$ ctags -R -f .tags * 2> /dev/null
:set tags += tags
```

v23

Múltiplos cursores

```
" programming
Plug 'mg979/vim-visual-multi'

:PlugInstall

# utilizando
posicione na palavra, CTRL+n até marcar todas, c,
digita a palavra nova e <ESC>
```

v24

Cores em Hexa

```
" programming
Plug 'ap/vim-css-color'

:PlugInstall
```

v25

Abrindo PDFs

precisa do pdftotext

```
" utils
Plug 'rhysd/open-pdf.vim'

:PlugInstall

# navegando
:Pdf <path> - converte, insere no cache a abre o texto
```

v26

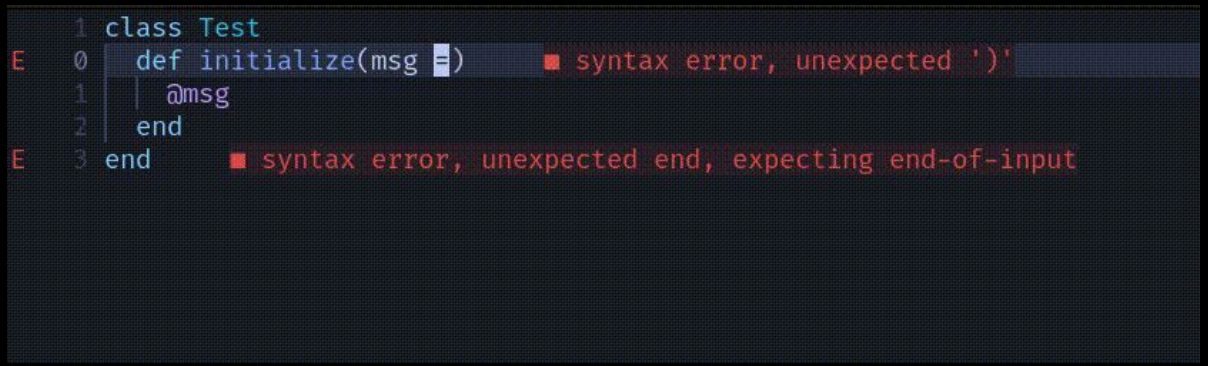
Convertendo underscore e camel

```
" utils
Plug 'tpope/vim-abolish'

:PlugInstall

# utilizando
crm - converte em CamelCase
crs - converte em snake_case
```

v28

Rodando linters

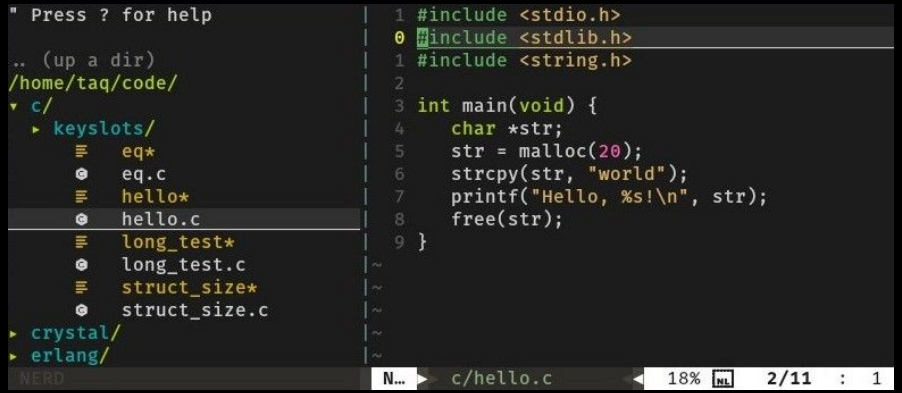
```
" utils
Plug 'dense-analysis/ale'

:PlugInstall
```



v29

Bônus: ícones

```
" instale uma NerdFont
https://github.com/ryanoasis/nerd-fonts

Plug 'ryanoasis/vim-devicons'

:PlugInstall
```



Obrigado!

Thank you!

Gracias!

Arigatô !

спасибо !

Twitter: @taq

E-mail: taq@eustaquiorangel.com

MACHINE HEAD

https://www.youtube.com/watch?v=i6F1x49h8uo